ANALISA PERBANDINGAN PROTOKOL MQTT DENGAN HTTP PADA IOT PLATFORM PATRIOT

COMPARISON ANALYSIS BETWEEN MQTT AND HTTP PROTOCOL IN PATRIOT IOT PLATFORM

Nindithia Putri Windryani[1], Dr. Nyoman Bogi A. K, S.T., MSEE.[2], Ratna Mayasari, S.T., M.T[3]
[1,2,3] Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom
[1] **ndiputri@students.telkomuniversity.ac.id** , [2] **aditya@telkomuniversity.co.id**,
[3]**ratnamayasari@telkomuniversity.ac.id**

Abstrak

Perkembangan teknologi IoT sedang berkembang dengan pesat di era globalisasi saat ini dan memberikan banyak manfaat serta kemajuan dalam berbagai aspek kehidupan manusia khususnya di bidang teknologi. Untuk menghubungkan berbagai perangkat elektronik, IoT membutuhkan suatu wadah yang disebut IoT Platform. IoT Platform Patriot dari penelitian sebelumnya yang digunakan pada tugas akhir ini masih dalam tahap pengembangan di beberapa aspek. Patriot Platform saat ini hanya support protokol HTTP yang masih memiliki beberapa kekurangan yaitu pemakaian bandwidth yang cukup besar, ukuran paket yang besar sehingga tidak reliable untuk berjalan pada sistem yang memiliki bandwidth rendah atau latency yang tinggi.

Berdasarkan kekurangan protokol HTTP tersebut, implementasi protokol MQTT server sangat dibutuhkan untuk menunjang pengembangan IoT Platform. MQTT merupakan protokol komunikasi yang sangat sederhana dan ringan. Protokol MQTT juga didesain untuk alat berkemampuan terbatas, bandwidth yang rendah, latency yang tinggi dan jaringan yang kurang dapat diandalkan.

Hasil pengujian QoS dengan parameter delay diperoleh rata-rata delay MQTT QoS 0 sebesar 0,0017s, QoS 1 sebesar 0,0628805s, MQTT QoS 2 sebesar 0,16987s dan HTTP sebesar 0,124591s. Packet loss yang di dapatkan sebesar 0% untuk MQTT QoS 1, QoS 2 dan HTTP sedangkan QoS 0 sebesar 13,3333%, nilai throughput protokol MQTT lebih rendah +/-324,7943 Bytes/s dibandingkan protokol HTTP sehingga protokol MQTT dapat lebih reliable berjalan pada keadaan bandwidth rendah atau latency tinggi dibandingkan protokol HTTP.

Kata kunci : MQTT, IOT Platform, IOT, Publish, Subscribe, Broker

Abstract

The development of the IoT technology is growing rapidly in the current era of globalization and provides many benefits and advances in various aspects of human life, especially in the field of technology. To connect various electronic devices, IoT requires a container called an IoT Platform. The Patriot Platform from previous research used in this final project is still in the development stage in several aspects. Patriot Platform currently only supports HTTP protocol which still has some disadvantages, namely the use of bandwidth is quite large, large packet size so it is not reliable to run on systems that have low bandwidth or high latency.

Based on the lack of the HTTP protocol, the implementation of the MQTT server protocol is needed to support the development of IoT Platform. MQTT is a very simple and lightweight communication protocol. The MQTT protocol is also designed for limited-capable devices, low bandwidth, high latency and less reliable networks.

The results of QoS testing with delay parameters obtained an average HTTP delay of 0.12340s higher than MQTT QoS 0, lower 0.05390s compared to MQTT QoS 1 and lower 0.227842s compared to MQTT QoS 2, packet loss at 0% for MQTT and HTTP, the MQTT protocol throughput is lower +/- 8.094 Kbit / s compared to the HTTP protocol so the MQTT protocol is more reliable in low bandwidth or high latency compared to the HTTP.

Keywords: MQTT, IOT Platform, IOT, Publish, Subscribe, Broker

1. Pendahuluan

Perkembangan Teknologi Internet of Things (IoT) sedang berkembang dengan sangat pesat di era globalisasi saat ini dan memberikan banyak manfaat serta kemajuan dalam berbagai aspek kehidupan manusia khususnya di bidang teknologi. Pada dasarnya teknologi Internet of Things (IoT) merupakan teknologi yang memungkinkan benda-benda yang ada disekitar kita terhubung ke jaringan internet. Dalam pengimplementasiannya untuk menghubungkan berbagai device dan perangkat elektronik IoT membutuhkan suatu wadah yang disebut IoT

Platform. IoT platform merupakan software berbasis web server yang berfungsi sebagai penghubung antara sensor, device, embedded system dan berbagai layanan lainnya agar dapat terhubung ke server. IoT Platform juga memiliki fungsi sebagai alternative yang mampu mendukung komunikasi berbagai perangkat IoT dan menjamin validitas perangkat yang mengirimkan data [1].

Saat ini IoT platform dari penelitian sebelumnya yaitu IoT platform Patriot masih dalam tahap pengembangan di berbagai aspek. Salah satunya, masih menggunakan protokol HTTP sebagai sarana komunikasi dua arah antar server. Pada protokol HTTP saat url di eksekusi respons dari server langsung diterima, di mana server tersebut dapat berupa plain text atau berformat JSON. Protokol HTTP memiliki beberapa kekurangan yaitu pemakaian bandwidth yang besar, ukuran paket yang besar sehingga kurang reliable untuk berjalan pada sistem yang memiliki bandwidth rendah atau latency tinggi. Maka dari itu, pengimplementasian protokol MQTT (Message Queuing Telemetry Transport) dapat menjadi salah satu solusi dari permasalahan tersebut. Berikut penelitian sebelumnya terkait implementasi protokol MQTT pada web server yaitu, "Rancang Bangun IoT Cloud Platform Berbasis Protokol Komunikasi MQTT" [1], "Implementasi MQTT (Message Queuing Telemetry Transport) pada Sistem Monitoring Jaringan berbasis SNMP (Simple Network Management Protocol)" [4] dan IoT Platform yang sedang dikembangkan merupakan tugas akhir Otsavianto Rukmanda yaitu, "Implementasi Metode Link Aggregation dengan Mode Balance-RR pada Operasional Server Berbasis IoT Platform" [7].

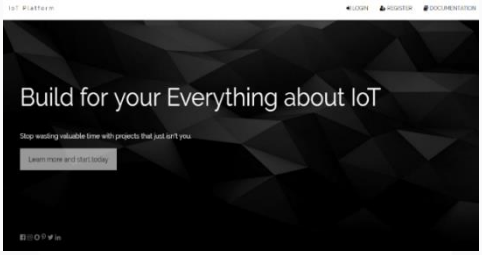
MQTT merupakan protokol komunikasi yang dirancang khusus untuk mendukung komunikasi "machine to machine" yaitu protokol dapat berkomunikasi dengan device/perangkat yang tidak memiliki alamat khusus [6]. MQTT merupakan protokol komunikasi yang sangat sederhana dan ringan. Protokol MQTT juga memiliki fungsi publish dan subscribe yang dimana dapat digunakan untuk komunikasi 2 arah yaitu dapat berkomunikasi antar server maupun dengan device lain [1][4]. Protokol MQTT juga di desain untuk alat berkemampuan terbatas, Bandwidth rendah, latency yang tinggi dan jaringan yang kurang dapat diandalkan. Tujuan dari pengimplementasian protokol MQTT Server adalah dapat meminimalisir penggunaan Bandwidth jaringan dan kebutuhan sumber daya pada perangkat sehingga sangat solutif dalam pengembangan IoT Platform [1][5].

2. Dasar Teori

2.1 IoT Platform

IoT Platform merupakan software berbasis web server yang berfungsi sebagai penghubung antara sensor, device dan berbagai layanan lainnya agar dapat terhubung ke server. Dimana hasil dari request oleh sensor, device, dan layanan lainnya akan di kirimkan ke server dan kemudian server akan merespon kembali sesuai dengan request yang telah diterima.

Pada Gambar 1. berikut merupakan tampilan platform IoT dari penelitian sebelumnya yang sedang dikembangkan.



Gambar 1. Tampilan IoT Platform [2].

2.2 MQTT (Message Queue Telemetry Transport)

MQTT (Message Queue Telemetry Transport) merupakan protokol yang berjalan diatas stack TCP/IP dan mempunyai ukuran paket data dengan low overhead yang kecil (minimum 2 bytes) sehingga berefek pada konsumsi catu daya yang juga cukup kecil, ringan dan sederhana [9]. Protokol MQTT juga didesain untuk alat-alat yang memiliki bandwidth rendah, latency atau keterlambatan yang tinggi, serta jaringan yang kurang dapat diandalkan dan perangkat yang memiliki kemampuan terbatas [9][10]. Prinsip kerja protokol MQTT adalah meminimalkan penggunaan bandwidth suatu network, kebutuhan sumber daya perangkat dan memastikan keandalan dalam pengiriman data [5].

Terdapat 3 level Quality of Service (QoS) dalam penyampaian pesannya, yaitu: [11]

a. QoS 0 (At Most Once): Pesan dikirimkan dengan upaya terbaik dari TCP/IP. Pesan hanya dapat dikirimkan satu kali atau tidak sama sekali.

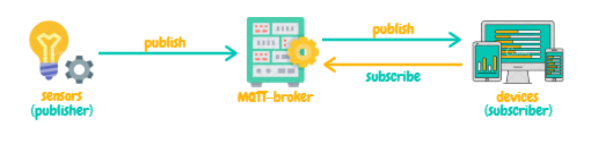
b. QoS 1 (At Least Once): Menjamin pesan akan dikirimkan setidaknya satu kali.

c. QoS 2 (Exactly Once): Pesan pasti tersampaikan dengan tepat dalam 1 kali.

2.3 Arsitektur Komunikasi Publish and Subscribe

MQTT adalah protokol komunikasi dengan arsitektur komunikasi publish/subscribe berbasis topik. Prinsip dari

arsitektur komunikasi model publish/subscribe adalah komponen-komponen yang menginginkan beberapa informasi dengan cara mendaftarkan topik yang diinginkan. Topik berfungsi sebagai filter untuk broker dalam pengiriman pesan ke tiap klien yang terhubung atau dengan kata lain topik berperan sebagai kanal bagi klien untuk melakukan subscribe [6].



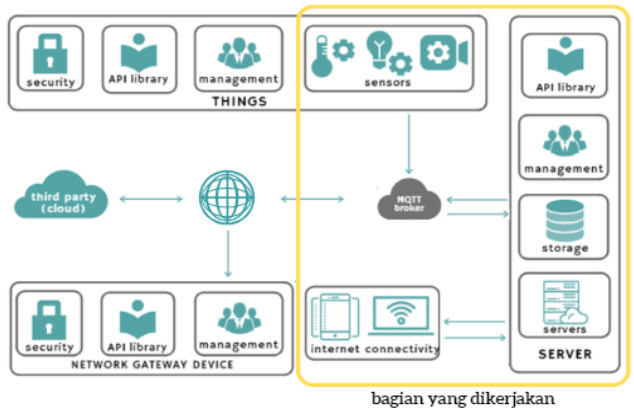
Gambar 2. Arsitektur Komunikasi Publish/Subscribe [1].

2.4 MQTT Broker Eclipse Mosquitto

Protokol MQTT memiliki broker sebagai pusat pertukaran informasi antara publisher dengan subscriber. MQTT Server / Broker menerima informasi yang telah didapat oleh publisher dengan melakukan inisiasi topik. Topik berfungsi sebagai alamat tujuan yang dikirim melalui MQTT Server atau Broker dari publisher sehingga yang dapat menerima informasi tersebut hanya subscriber yang mengetahui topik yang digunakan [3][13]. Beberapa contoh MQTT Broker antara lain Mosquitto, Mosca, Emqttd. MQTT Broker yang digunakan oleh penulis pada tugas akhir ini adalah Eclipse Mosquitto yaitu lightweight open source message broker yang mengimplementasikan protokol MQTT versions 3.1 dan 3.1.1 untuk mengirim dan menerima messages, khususnya dengan perangkat IOT (Internet Of Things) [8].

3. Model Sistem dan Perancangan

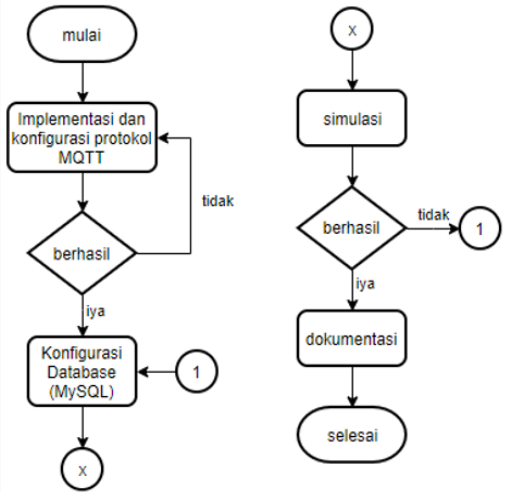
3.1 Blok Sistem IoT Platform



Gambar 3. Blok Sistem IoT *Platform*

Dalam blok sistem IoT Platform terdapat 3 bagian utama yaitu Things, Network gateway devices, dan Server. Seperti yang bisa dilihat pada Gambar 3 di Tugas Akhir ini bagian yang dikerjakan oleh penulis adalah Things, Server dan Internet Connectivty.

3.2 Proses Kerja Protokol MQTT pada IoT Platform

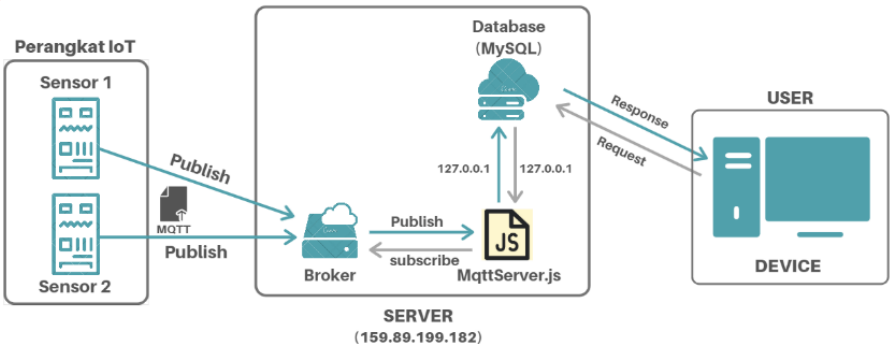


Gambar 4. Alur Kerja Protokol MQTT pada IoT Platform

Proses kerja protokol MQTT pada IoT platform seperti pada Gambar 4. di bawah ini, tahap pertama adalah Melakukan implementasi dan konfigurasi protokol MQTT pada IoT *platform*, kemudian Membangun koneksi dengan database. Jika sudah terkoneksi dengan database, lakukan simulasi untuk mengetahui apakah protokol MQTT sudah berhasil diimplementasikan pada IoT *Platform.* Apabila sudah berhasil maka dilakukan dokumentasi dan pengambilan data dari hasil simulasi.

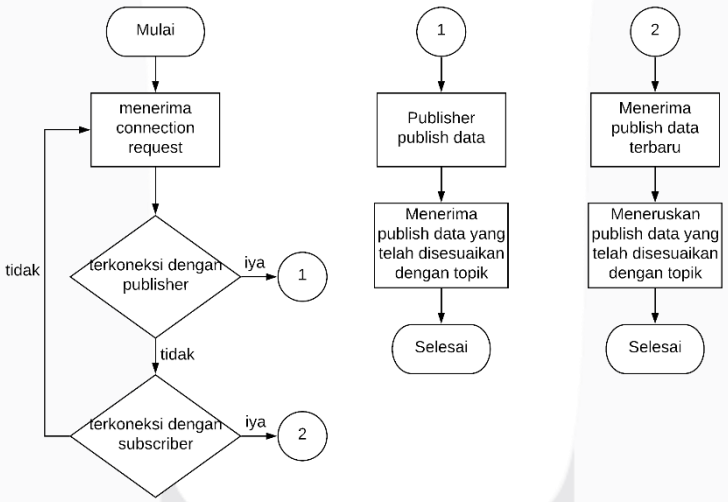
3. 3 Perancangan dan Proses Kerja Sistem

Tahap pertama perancangan dan proses kerja dari protokol MQTT adalah publisher mengirimkan data berupa topik, Api Token, nilai dan satuan menuju broker, kemudian broker menerima semua data yang dikirimkan oleh publisher. Lalu, subscriber melakukan request data menggunakan IP broker. Setelah itu, broker dan subscriber terkoneksi dan subscriber dapat menerima data publish yang dikirimkan oleh publisher melalui perantara broker. Subscriber akan mencocokkan payload data publish yang diterima dari publisher dengan database melalui spesifikasi berupa topik, access–key dan nilai sensor. Jika spesifikasi data sudah sesuai maka data tersebut dimasukkan ke dalam database. Terakhir, data tersebut dapat ditampilkan pada IoT platform.



Gambar 5. Proses Kerja Sistem MQTT

3.4 Perancangan Alur Kerja Broker

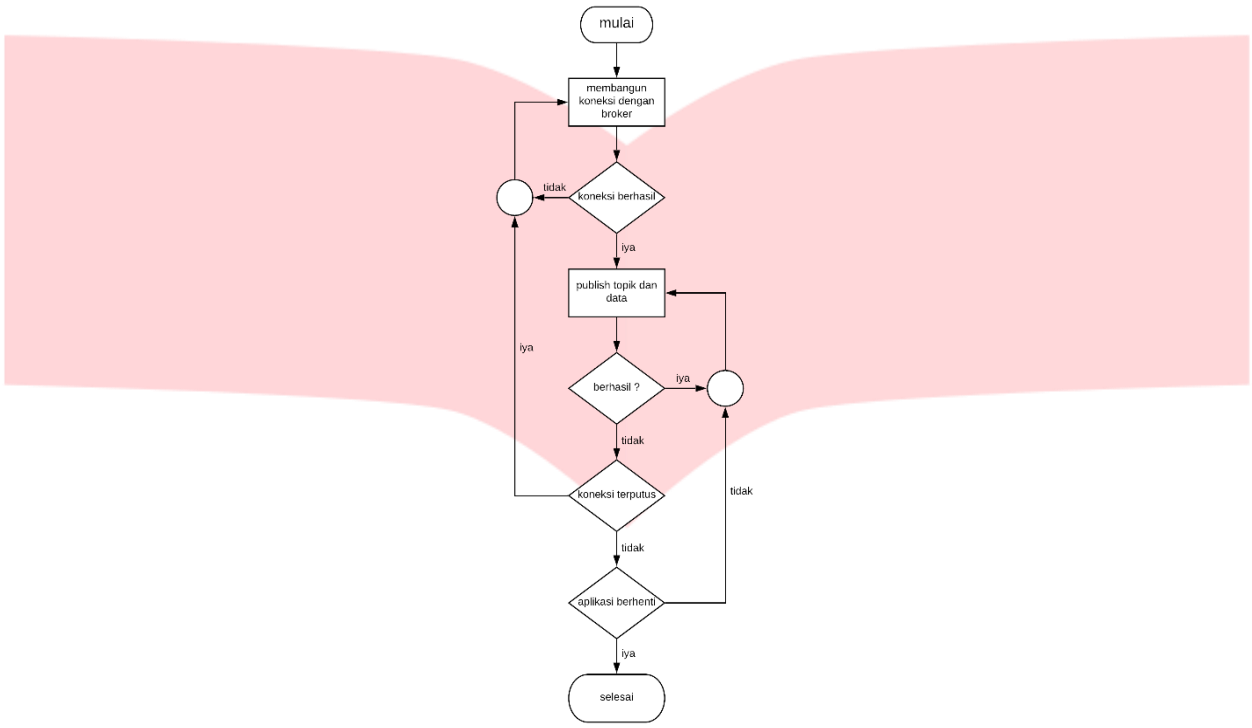


Gambar 6. Diagram Alir Broker

Pada tahap perancangan alur kerja broker terdapat proses menerima publish data dengan topik tertentu dari publisher. Lalu, proses meneruskan publish data ke subscriber yang melakukan subscribe data yang sama. Langkah pertama adalah fungsi MQTT Broker dijalankan. Kemudian, jika ada publisher mengirimkan publish data akan disimpan dan diterima oleh broker. Dan ketika subscriber melakukan subscribe topik yang sama dengan publisher, broker akan meneruskan data publish yang telah diterima ke subscriber. Gambar 6 menunjukkan diagram alir rancangan kerja broker pada arsitektur publish/subscribe.

3.5 Perancangan Alur Kerja Publisher

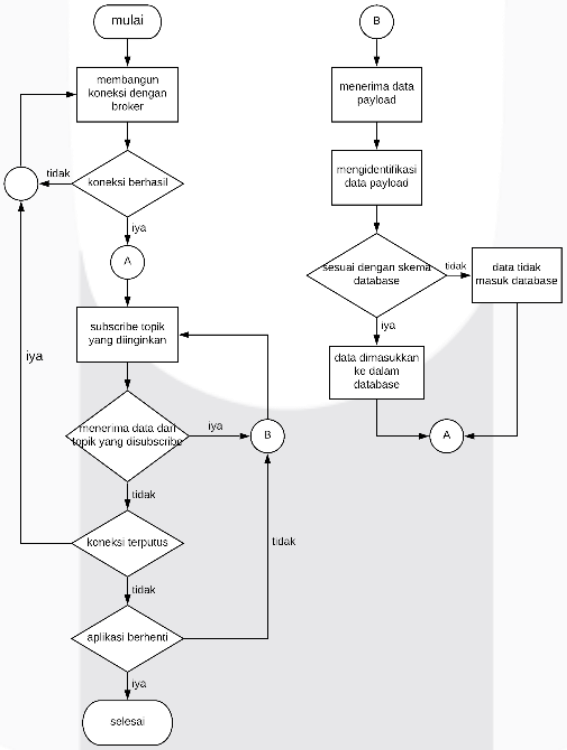
Pada tahap ini, dilakukan perancangan alur kerja publisher dari protokol komunikasi MQTT. Dimulai dengan publisher membangun koneksi dengan broker. Tahap selanjutnya, jika publisher berhasil membangun koneksi dengan broker maka publisher dapat melakukan publish data dengan topik yang telah ditentukan menuju broker kemudian broker akan mendeteksi adanya permintaan publish data dari publisher. berikut merupakan diagram alir perancangan alur kerja publisher pada protokol komunikasi MQTT.



Gambar.7 Diagram Alir Publisher

3.6 Perancangan Alur Kerja Subscriber

Langkah pertama yang harus dilakukan oleh subscriber adalah membangun koneksi dengan broker agar dapat melakukan subscribe data pada topik tertentu. Pertama, broker akan mendeteksi permintaan koneksi dari subscriber, kemudian setelah subscriber berhasil terkoneksi dengan broker maka pada tahap selanjutnya subscriber dapat melakukan subscribe data dengan topik dan Access-key yang sudah sesuai dengan database kemudian data akan dimasukkan ke dalam database. Pada Gambar 8. menunjukkan diagram alir proses kerja subscriber



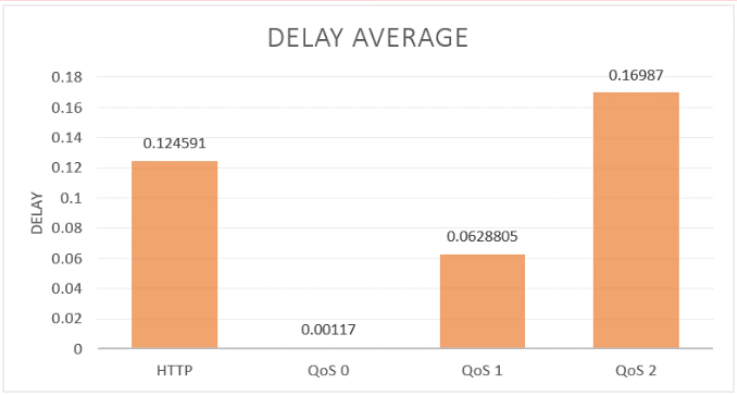
Gambar 8. Alur Kerja Subscriber

4. Pengujian dan Analisis

Berikut merupakan beberapa analisis dari hasil pengujian dengan parameter uji QoS:

4.1 Delay

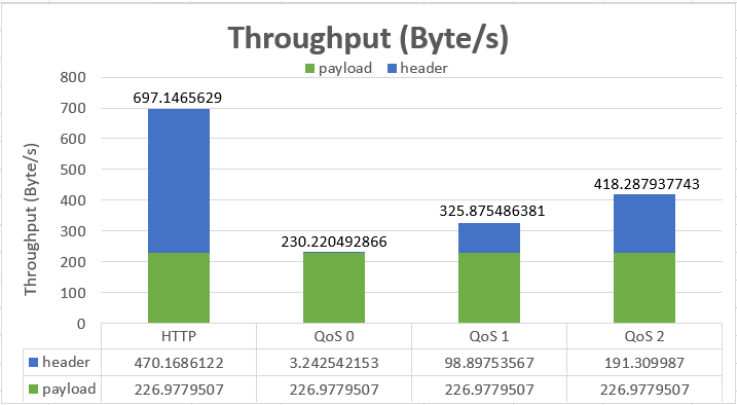
Pada Gambar 9. diperoleh hasil rata-rata delay dari pengujian yang sudah dilakukan sebelumnya. Hasil rata-rata delay protokol MQTT QoS 0 sebesar 0.00117s dengan deviasi 0.00048s, QoS 1 sebesar 0.0628805s dengan deviasi 0.01796s, QoS 2 sebesar 0.16987s dengan deviasi 0.05619s, dan protokol HTTP sebesar 0.124591s dengan deviasi 0.01087s. Berdasarkan hasil tersebut diketahui bahwa MQTT QoS 0 dan QoS 1 memiliki rata-rata delay yang lebih baik dibandingkan HTTP. Sedangkan MQTT QoS 2 memiliki rata-rata delay yang lebih tinggi 0.045s dibandingkan HTTP, disebabkan sistem pengiriman data MQTT QoS 2 yang lebih kompleks. Berdasarkan parameter ITU-T rata-rata delay MQTT QoS 0, 1 dan 2 tergolong baik.



Gambar 9. Grafik Delay

4.2 Pengujian Throughput

Berdasarkan hasil pengujian, diperoleh hasil throughput dengan grafik seperti pada Gambar 10. Grafik tersebut menunjukkan bahwa protokol HTTP memiliki nilai throughput paling tinggi yaitu sebesar 697,1465629 Bytes/s dan MQTT QoS 0 memiliki nilai throughput paling rendah yaitu sebesar 230,2204929 Bytes/s, MQTT QoS 1 sebesar 325,8754864 Bytes/s dan MQTT QoS 2 sebesar 418,2879377 Bytes/s. Kemudian Gambar 11. Berikut menunjukkan grafik persentase pemakaian bandwidth protokol MQTT dan HTTP. Dapat dilihat pada grafik tersebut pemakaian bandwidth protokol MQTT (QoS 0, QoS 1, QoS 2) lebih kecil dibandingkan protokol HTTP. Sehingga, dapat disimpulkan bahwa walaupun throughput MQTT lebih kecil dibandingkan protokol HTTP, jumlah bandwidth yang digunakan protokol MQTT lebih sedikit dibandingkan protokol HTTP dikarenakan packet size protokol MQTT juga relatif lebih kecil dibandingkan protokol HTTP. Sehingga pada kondisi suatu sistem memiliki bandwidth yang rendah atau latency yang tinggi protokol MQTT lebih reliable untuk digunakan dibandingkan protokol HTTP.

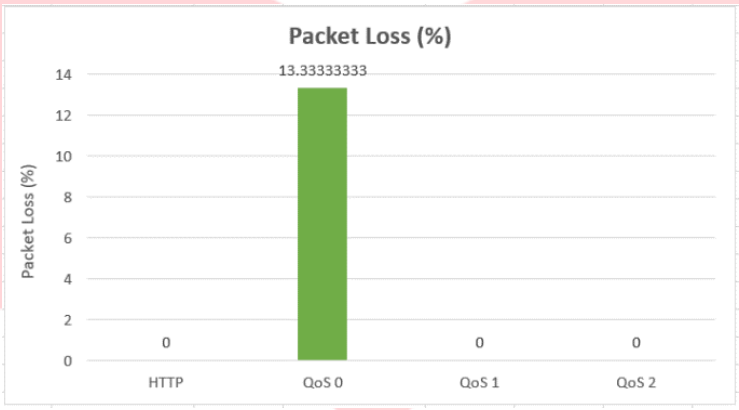


Gambar 10. Grafik Throughput

4.3 Pengujian Packet Loss

Berdasarkan hasil pengujian packet loss sebelumnya diperoleh hasil packet loss protokol HTTP, MQTT QoS 1 dan QoS 2 sebesar 0% sedangkan MQTT QoS 0 sebesar 13,3333%. Hal ini dikarenakan protokol MQTT QoS 0 bersifat fire and forget yang dimana jika data yang dikirimkan tidak berhasil diterima oleh penerima maka tidak ada pemberitahuan untuk dilakukannya pengiriman ulang sehingga data hanya akan berhasil dikirimkan sekali atau tidak sama sekali. Sedangkan, sistem pengiriman data protokol HTTP, MQTT QoS 1 dan QoS 2 memastikan bahwa data yang dikirimkan pasti dapat sampai ke tujuan minimal satu kali untuk QoS 1 dan tepat satu kali untuk QoS 2. Pengujian dilakukan dalam kondisi lingkungan yang memiliki bandwidth yang rendah (10 KBytes/s) dan

latency yang tinggi sehingga packet loss pada QoS 0 dapat terjadi. Berdasarkan tabel parameter packet loss pada bab sebelumnya packet loss protokol MQTT QoS 1 dan QoS 2 tergolong sangat bagus dan MQTT QoS 0 tergolong bagus karena masih berada dibawah 15%. Gambar 11. berikut merupakan grafik hasil pengujian packet loss.



Gambar 11. Grafik Packet Loss

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasakarkan hasil pengujian dan implementasi yang telah dilakukan sebelumnya, penulis dapat mengambil kesimpulan bahwa:

1. Protokol MQTT dapat di implementasikan pada server IoT Platform
2. Hasil pengujian performansi implementasi protokol MQTT pada IoT Platform dengan parameter delay diperoleh nilai rata-rata delay saat pengiriman data menggunakan protokol MQTT sebesar 0,0017s untuk QoS 0, sebesar 0,0628805s untuk QoS 1, dan 0,16987s untuk QoS 2 sedangkan protokol HTTP sebesar 0,12459s. Yang dimana delay rata-rata QoS 0 dan QoS 1 lebih kecil dibandingkan protokol HTTP sedangkan QoS 2 lebih besar 0,045279s dibandingkan HTTP disebabkan sistem pengiriman QoS 2 yang lebih complex.
3. Pengujian performansi implementasi protokol MQTT pada IoT Platform dengan parameter packet loss pada kondisi lingkungan dengan bandwidth yang kecil dan latency yang tinggi menggunakan protokol HTTP dan MQTT diperoleh hasil packet loss sebesar 0% untuk protokol MQTT QoS 1, QoS 2 dan HTTP. Packet loss sebesar 13,33333% untuk protokol MQTT dengan QoS 0 dikarenakan sistem pengiriman data pada MQTT QoS 0, jika data gagal dikirimkan, tidak ada pengiriman kembali sehingga packet loss dapat terjadi.
4. Pengujian performansi implementasi protokol MQTT pada IoT Platform dengan parameter throughput diperoleh hasil pada protokol MQTT QoS 0 dengan nilai 230,220492866 Bytes/s, QoS 1 dengan nilai 325,875486381 Bytes/s, QoS 2 dengan nilai 418,287937743 Bytes/s dan protokol HTTP dengan nilai 697,1465629 Bytes/s. Protokol MQTT memiliki nilai throughput yang lebih rendah dibandingkan protokol HTTP disebabkan packet size protokol MQTT lebih kecil dibandingkan protokol HTTP. Sehingga protokol MQTT dapat meminimalisir penggunaan bandwidth jaringan.
5. MQTT dapat diimplementasikan pada platform IoT Patriot dengan melakukan instalasi dan konfigurasi broker di server IoT platform Patriot. Melakukan konfigurasi dan membangun koneksi antara publisher dan subscriber dengan broker.

5.2 Saran

1. Pengujian performansi Protokol MQTT di uji dengan lebih banyak parameter.
2. Pengujian dilakukan dengan sensor yang lebih bervariasi.
3. Client dapat menerima data, menampilkan data dan menjadi subscriber dalam komunikasi publish/subscribe.

Daftar Pustaka:

[1] M. Habibi, Wildan, A. Bhawiyuga, and A. Basuki, “Rancang Bangun IoT Cloud Platform Berbasis Protokol Komunikasi MQTT,” Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 2(2), p.479-485. 2017.

[2] R. Otsavianto,”Implementasi Metode Link Aggregation Dengan Mode Balance-RR pada Operational Server Berbasis IOT Platform”, Telkom University, 2018.

[3] MQTT.org, 2016. MQTT. Available at: http://mqtt.org/faq [Diakses 15 November].

[4] HIVE MQ, “MQTT Essentials Part 1,” Available at: https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt. [Diakses 15 November].

[5] Eclipse Mosquitto, "An Open Source MQTT Broker". Available at: https://mosquitto.org/ [accessed 19 Oktober 2018].

[6] A. Putra and F. Ramadoni, "Usung Kemudahan Setup Layanan IoT, myDevices Rilis Integrasi," TeknoJurnal. Available at: https://teknojurnal.com/usung-kemudahan-setup-layanan-iot-mydevicesrilis-integrasi-cayenne-dengan-arduino-dengan-fitur-drag-drop/. 2016. [Accessed 3 Oktober 2018].

[7] J. Chen, and W. Cheng, "Analysis of Web Traffic Based on HTTP Protocol," IEEE International Conference in Software, Telecommunications and Computer Networks, 2016.

[8] A. Segara, Pandu, R. Primananda, and A. S. Rizqika, "Implementasi MQTT (Message Queuing Telemetry Transport) pada Sistem Monitoring Jaringan berbasis SNMP (Simple Network Management Protocol)," Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 2(2), p.695-702. 2018.

[9] R. K. Kodali and K. S. Mahesh, "A low cost implementation of MQTT using ESP8266," IEEE 2nd International Conference on Contemporary Computing and Informatics (IC3I), 2016.

[10] M. Yue, Y. Ruiyang, S. Jianwei, and Y. Kaifeng,"A MQTT Protocol Message Push Server Based on RocketMQ," IEEE 10th International Conference on Intelligent Computation Technology and Automation (ICICTA), 2017.

[11] S. Shin, K. Kobara, C. Chuang, and H. Weicheng,"A Security framework for MQTT," IEEE Conference on Communications and Network Security (CNS), 2016.

[12] R. K. Kodali, V. S. Gorantla,"Weather Tracking System using MQTT and SQLite," IEEE International Conference and Applied and Theoritical Computing and Communication Technology(iCATccT), 2017.

[13] R. A. Light, "Mosquitto: server and client implementation of the MQTT protocol," The Journal of Open Source Software, vol. 2, no. 13, May 2017.